

Setting up a Simple Database Infrastructure using Amazon RDS

Introduction

The goal of this assignment was to create a SQL database using the AWS console and establish a connection to it from a local machine using MySQL Workbench. This involved setting up an RDS instance in AWS, selecting the appropriate configurations such as database engine and credentials, and ensuring that the database was accessible from our local environment. Through this process, we explored how cloud-hosted databases can be effectively managed and accessed using familiar local tools, making it easier to integrate cloud services into everyday development tasks.



Group Members

- Esther Acheampong
- Eugene Essuman
- Grace Wiredu
- Habiba Adam Salisu
- Benedicta Maayuumle Djangmah
- Bernard Solodzi
- Osman Abdul
- Michael Ohene Aboagye
- Nick Anderson
- Rachel Atia
- Vera Sekyere
- Azal Innocentia
- Akoto Nana Yaw Appiah
- Rhoda Kudor
- Benedicta Opoku-Amankwaah
- Jessica Azumah
- Rose Anyama Owusu
- Mohammed Razak Mikdar
- Clement Owusu Bempah

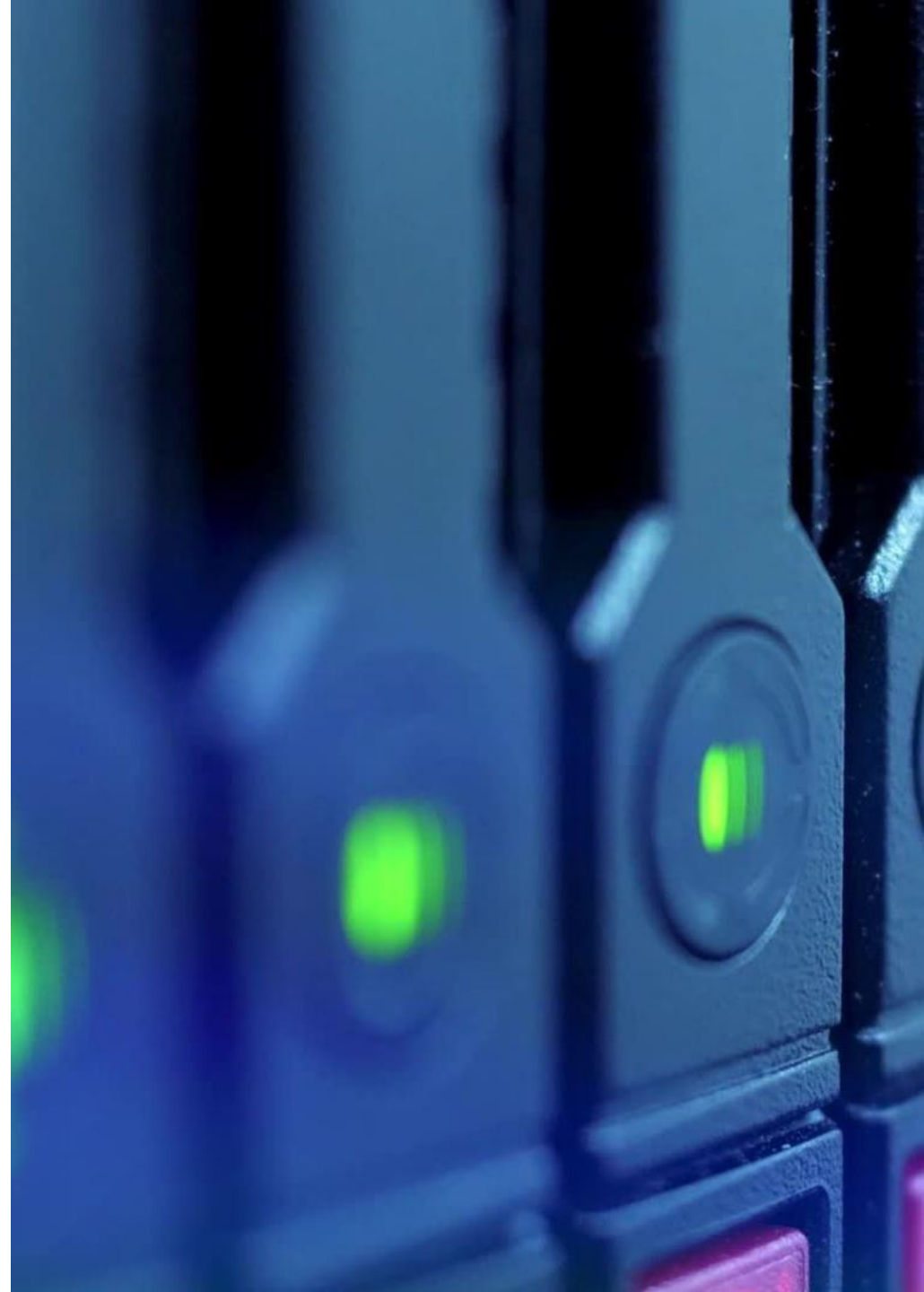
Step 1 to 4 - Launch RDS on AWS Console

Process:

- Logged into AWS Management Console.
- Navigated to **Amazon RDS > Databases > Create Database > Selected A Region(Virginia)**

Lessons Learned:

- AWS RDS supports multiple database engines (MySQL, PostgreSQL, etc.).
- GUI-based deployment makes setup easy for beginners and professionals.



Step 4 to 5 - Creation of Database Instance named: room2-database

The screenshot shows the AWS Management Console interface for creating a new database instance. The browser address bar indicates the URL: <https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:>. The console header shows the AWS logo, a search bar, and the user's name, Habiba Adam Salisu, in the United States (N. Virginia) region.

The main content area is titled "Aurora and RDS > Create database". It is divided into two main sections: "Settings" and "MySQL".

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 63 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

☐ **Managed in AWS Secrets Manager - *most secure***
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

☒ **Self managed**
Create your own password or have RDS create a password that you manage.

☐ **Auto generate password**
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

MySQL [>](#)

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross region.

The bottom of the console shows a footer with "© 2025, Amazon Web Services, Inc. or its affiliates." and links for "Privacy", "Terms", and "Cookie preferences". The Windows taskbar at the bottom shows the date and time as 3:37 pm on 10/06/2025.

Step 6 and 7

DB Instance Settings: Provide a unique DB instance identifier, a master username, and a secure password (8-41 printable ASCII characters, excluding /, ", and @).

Instance Class: Choose db.t2.micro (1 vCPU, 1 GiB RAM) for basic usage.

Storage Configuration: Use **General Purpose (SSD)** with **20 GB allocated storage** (scalable up to 64 TB); autoscaling is not enabled in this tutorial.

Multi-AZ Deployment: Optional feature that adds high availability with a standby replica in another Availability Zone, but incurs additional cost.

See in Fig. 2

View of Step 6 and 7

Aurora and RDS

> Create database

Burstable classes (includes t classes)

db.t4g.micro

2 vCPUs 1 GiB RAM Network: Up to 2,085 Mbps

Storage

Storage type

Info

Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp2)

Baseline performance determined by volume size

Allocated storage

Info

20

GiB

Allocated storage value must be 20 GiB to 6,144 GiB

► Additional storage configuration

Connectivity

Info

Compute resource

Fig. 2

Details of Database: room2-database created

The screenshot displays the AWS Management Console interface for the 'room2-database' instance. The browser address bar shows the URL: `https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=room2-database;is-cluster=false`. The console header includes the AWS logo, a search bar, and navigation links for 'United States (N. Virginia)' and the user 'Habiba Adam Salisu'.

The left sidebar shows the 'Aurora and RDS' navigation menu with options like Dashboard, Databases, Query editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, and Zero-ETL integrations.

The main content area features a green success message: 'Successfully created database room2-database'. Below this, the database name 'room2-database' is displayed with 'Modify' and 'Actions' buttons. The 'Summary' section provides the following details:

DB identifier	Status	Role	Engine
room2-database	⌚ Backing-up	Instance	MySQL Community

CPU	Class	Current activity	Region & AZ
53.21%	db.t4g.micro	0 Connections	us-east-1b

Below the summary, there are tabs for 'Connectivity & security' (selected), 'Monitoring', 'Logs & events', 'Configuration', and 'Zero-ETL integrations'. The 'Connectivity & security' section is currently visible.

The footer of the console shows 'CloudShell', 'Feedback', and copyright information for Amazon Web Services, Inc. or its affiliates. The Windows taskbar at the bottom indicates a temperature of 30°C, a search bar, and the system time of 3:53 pm on 10/06/2025.

Copying of Necessary Details For Local Connection

The screenshot displays the AWS Management Console interface for an Amazon RDS database instance. The browser address bar shows the URL: `https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#database:id=room2-database;is-cluster=false`. The console navigation pane on the left lists various RDS services, with 'Databases' selected. The main content area is titled 'room2-database' and features a tabbed interface with 'Connectivity & security' as the active tab. This tab is divided into three sections: 'Endpoint & port', 'Networking', and 'Security'. The 'Endpoint & port' section shows the endpoint `room2-database.cmxkqeks8mz5.us-east-1.rds.amazonaws.com` and port `3306`. The 'Networking' section lists the availability zone as `us-east-1b`, the VPC as `vpc-0652027dea046ba69`, and a list of subnets. The 'Security' section indicates that the VPC security group `room2group (sg-0b27772850af7e2ce)` is active, that the database is publicly accessible, and provides information about the certificate authority and its expiration date.

Endpoint & port

- Endpoint: `room2-database.cmxkqeks8mz5.us-east-1.rds.amazonaws.com`
- Port: `3306`

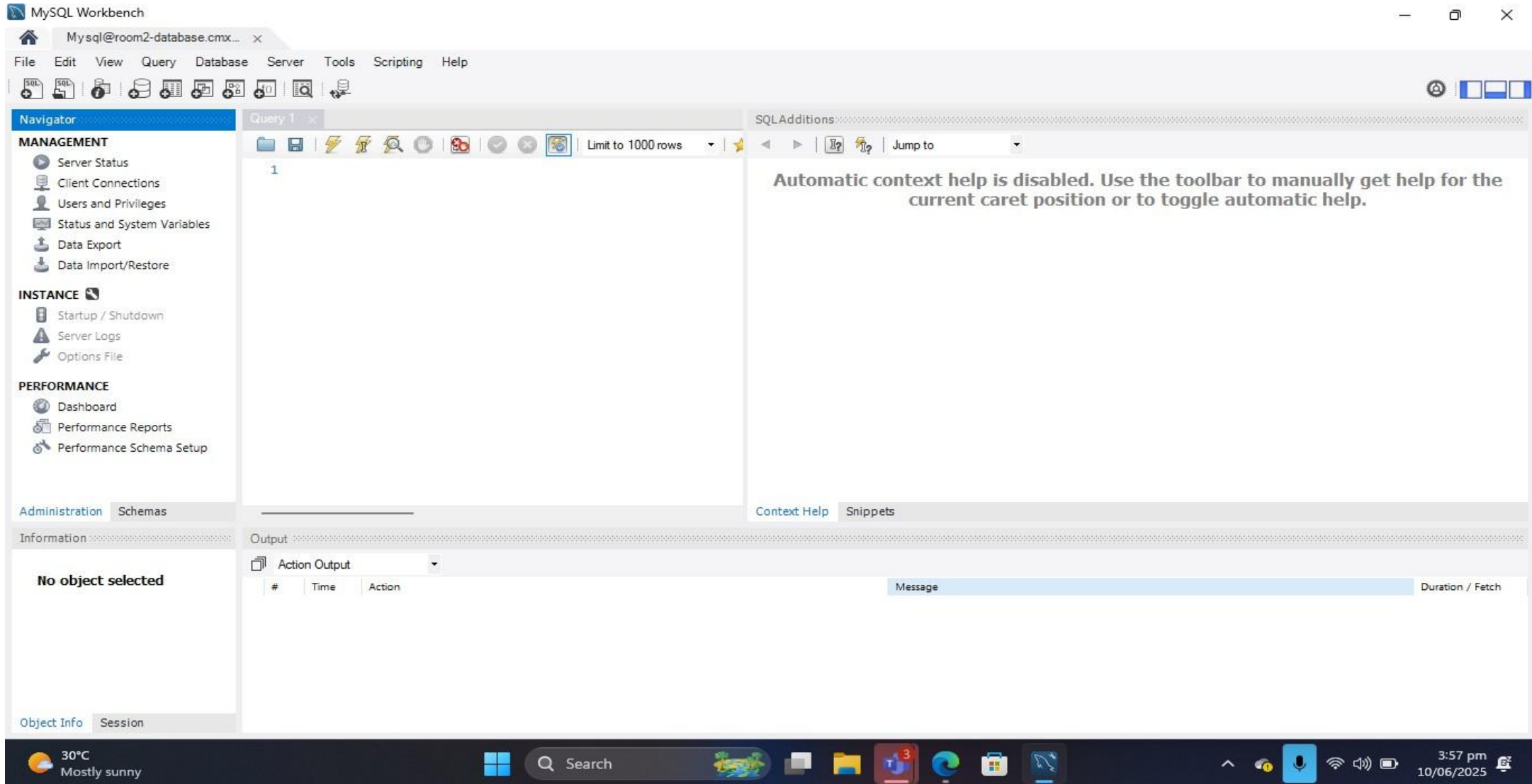
Networking

- Availability Zone: `us-east-1b`
- VPC: `vpc-0652027dea046ba69`
- Subnet group: `default-vpc-0652027dea046ba69`
- Subnets:
 - `subnet-012fe709c3ec98db5`
 - `subnet-0b6597384261bacfc`
 - `subnet-0fad5ce559d7fcfea`
 - `subnet-0f59eae72a40b9d3c`
 - `subnet-0f9d54eea2808dc8b`
 - `subnet-0d711efae585d622e`

Security

- VPC security groups: `room2group (sg-0b27772850af7e2ce)` (Active)
- Publicly accessible: Yes
- Certificate authority: `rds-ca-rsa2048-g1` (Info)
- Certificate authority date: May 25, 2061, 23:34 (UTC+00:00)
- DB instance certificate expiration date: June 10, 2026, 15:50 (UTC+00:00)

Launch MySQL Workbench



Query For Creating A Table in MySQL Workbench

```
CREATE TABLE `employee` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `name` varchar(255) DEFAULT NULL,  
  `email` varchar(255) DEFAULT NULL,  
  `city` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`id`) )
```

We used the script above to create a table in MySQL workbench

See in Fig. 3

Creating A Table With Details

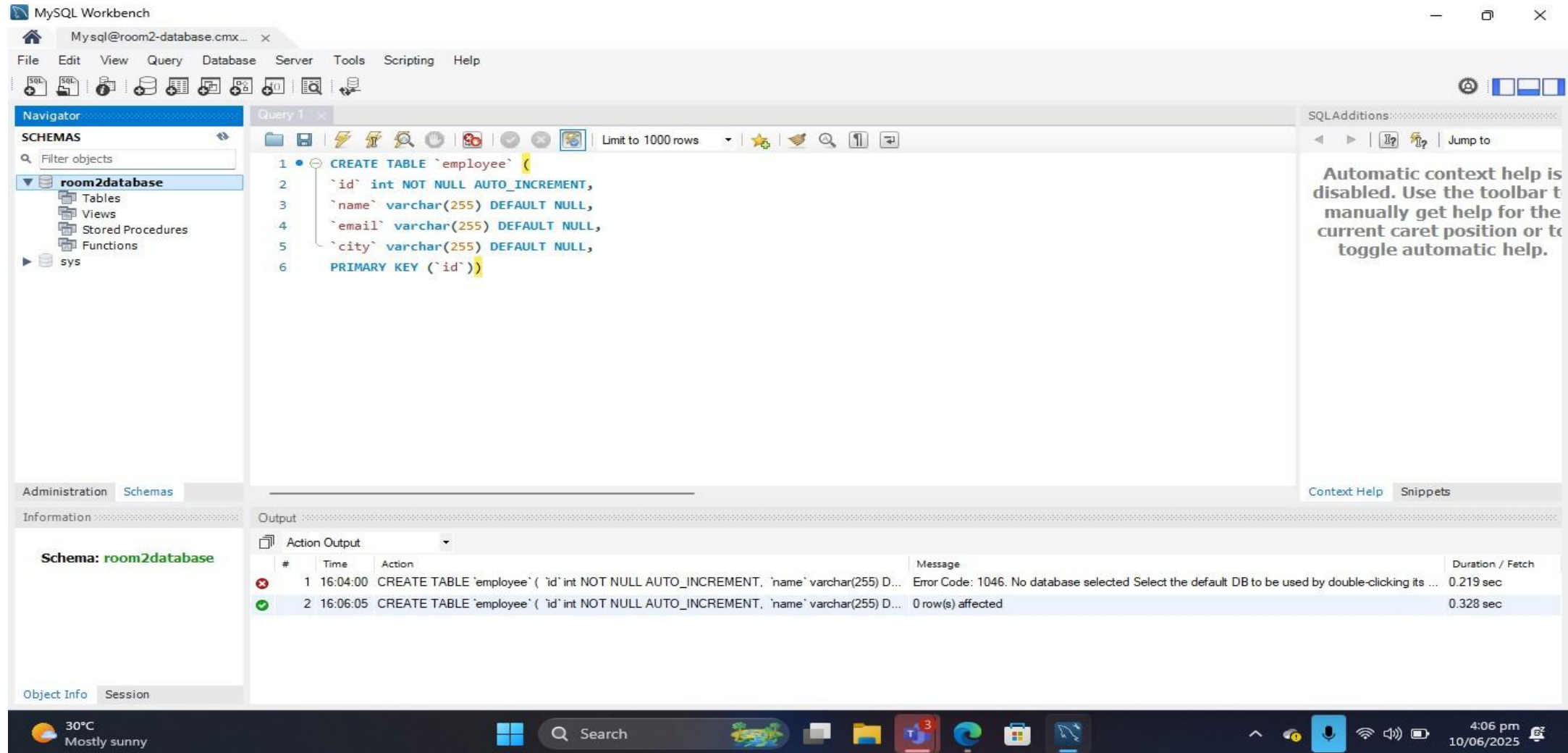


Fig. 3

Viewing of Data in "employee" table

The screenshot displays the MySQL Workbench interface. The 'Navigator' pane on the left shows the 'room2database' schema with tables, views, stored procedures, and functions. The 'Query Editor' shows a query: `select * from employee`. The 'Result Grid' displays the data from the 'employee' table. The 'Output' pane shows the execution log with four actions: creating the table, inserting data, and selecting data. The 'Schema: room2database' pane shows the schema details. The 'Object Info' pane shows the session information.

Result Grid

	id	name	email	city
1	Group two.	group.two@example.com	Kumasi	
2	Osman Abdul	osman.abdul@example.com	Accra	
3	rachel atia	rachel.atia@example.com	Takoradi	
4	vera Okyere	vera.okyere@example.com	Tamale	
5	dement owusu	dement.owusu@example.com	Cape Coast	
*	NULL	NULL	NULL	NULL

Output

#	Time	Action	Message	Duration / Fetch
1	16:04:00	CREATE TABLE 'employee' ('id' int NOT NULL AUTO_INCREMENT, 'name' varchar(255) ...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking it...	0.219 sec
2	16:06:05	CREATE TABLE 'employee' ('id' int NOT NULL AUTO_INCREMENT, 'name' varchar(255) ...	0 row(s) affected	0.328 sec
3	16:10:43	INSERT INTO 'employee' ('name', 'email', 'city') VALUES ('Group two.', 'group.two@example.com', 'Kumasi')	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.312 sec
4	16:12:19	select * from employee LIMIT 0, 1000	5 row(s) returned	0.219 sec / 0.000 sec

Conclusion and Summary

This project provided hands-on experience in deploying a managed MySQL database using Amazon RDS, connecting to it securely, and executing SQL operations through MySQL Workbench. Throughout the process, key AWS concepts such as instance provisioning, networking, security groups, automated backups, and resource cleanup were explored.

Key Lessons Learned:

- AWS RDS simplifies database setup by abstracting infrastructure management, making it ideal for scalable applications and learning environments.
- Security group configuration is critical for enabling external access; public access must be combined with correct IP and port settings.
- MySQL Workbench is a powerful GUI tool for connecting to and managing cloud databases.
- Testing with basic SQL scripts validates that the database is fully operational.
- Always clean up unused resources (like DB instances) to avoid unnecessary charges and maintain a tidy cloud environment.

Final Thought:

Mastering the end-to-end workflow of creating, connecting to, and managing RDS databases lays a solid foundation for more advanced cloud-native data applications and AWS certification readiness.